CHAPTER

5

USING GRAPHICAL SYSTEMS WITH LINUX

After reading this chapter and completing the exercises, you will be able to:

- ♦ Discuss concepts related to the X Window System
- ♦ Install and configure the X Window System
- ♦ Start up and use the most popular graphical interfaces on Linux
- ♦ Use a graphical login

In the previous chapter you learned basic commands for working with the Linux file system. You also learned how to manage software packages and make modifications to the Linux kernel by using kernel modules or by recompiling the Linux kernel. In addition, you learned how a Linux system is initialized through a series of scripts that are started by the `init` program.

In this chapter you will learn how to work with a graphical environment in Linux and how to configure access to graphics hardware. You will also practice using the most popular graphical interfaces available for Linux. In addition, you will learn how to set up a graphical login screen so that the users on a Linux system can work continuously in a graphical environment.

LEARNING ABOUT THE X WINDOW SYSTEM

When you installed Linux in Chapter 3, you attempted to configure the X Window System, the graphical system in Linux that is often called, simply, X. Because the graphical system can be challenging to configure, you may need the additional information provided in this chapter to successfully set up the graphical environment. Before exploring additional configuration details, however, you should take a moment to become familiar with the history of the X Window System.

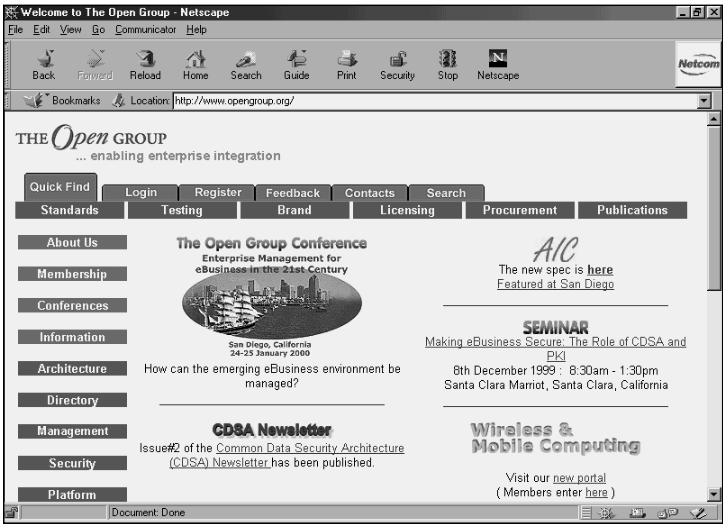
A Brief History of X

The rise of graphical environments on computer systems began in the early 1980s. The seeds of this trend were planted at the Palo Alto Research Center of Xerox Corporation, where researchers did the earliest work on graphical user interfaces (GUIs) and devices for interacting with them (including the mouse). But Xerox was not the company that made GUIs popular. That honor fell to three different groups, each working independently toward the goal of making its computer system easier to use. The first GUI to be made widely available was an early version of Windows from Microsoft—Windows 1.0. However, this version and the later version, Windows 2.0, were not functional enough to be anything more than technological curiosities. Conversely, the Apple Macintosh, which began shipping in 1984, made full use of GUI technology, such as menus and dialog boxes, all within a functional and stable environment. At Microsoft, developers continued working on versions of Windows until, by the 3.1 release, the product was functional and stable enough to gather a following in the market.

The third group working on graphical environments was made up of developers bent on creating a useful GUI for UNIX. At the time Windows was establishing its place in the market (the mid-1980s), UNIX had already been in widespread use for years. To make the operating system easier to use and to encourage the development of graphical standards, people at the Massachusetts Institute of Technology (MIT) and Digital Equipment Corporation (DEC, now part of Compaq Computer Corporation) began working together on a graphical environment for UNIX. This development process was initially dubbed **Project Athena**. The new graphical environment was eventually called the X Window System, with the assumption that *X* would be replaced with something more descriptive. But the X appellation stuck and is used to this day.

The X Window System, or X, as it is commonly known, was released as public domain software in 1985. This allowed many UNIX vendors to begin creating products based on X, and it rapidly became the default graphical system for the entire UNIX market. The GPL (the license under which Linux was released) was not developed by Richard Stallman of the Free Software Foundation until 1992, so X was released under a different legal arrangement. By placing the software in the public domain, the developers (MIT and DEC) gave up their copyright to the software, leaving all others free to create derivative works and copyright them. The result was a somewhat fragmented market, in which users could choose from various graphical systems based on X. Because the UNIX market was already fragmented and did not rely on the mass-market economies that are associated with computers today, the availability of many varieties of X was not considered problematic.

In 1988 the Open Software Foundation, or OSF, took over work on the development project (Athena) that created new versions of X. The not-for-profit OSF (now called the Open Group) continues to maintain X to this day. Because X is public domain software, however, the source code is available to anyone, much like the source code to the Linux kernel or the GNU utilities. Figure 5-1 shows the Open Group's Web site, at *www.opengroup.org*, where you can find more information about X. For additional background, visit *www.x.org*.


Welcome to The Open Group - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop Netscape
Netcom
Bookmarks Location: http://www.opengroup.org/
THE Open GROUP
... enabling enterprise integration
Quick Find Login Register Feedback Contacts Search
Standards Testing Brand Licensing Procurement Publications
About Us
Membership
Conferences
Information
Architecture
Directory
Management
Security
Platform
Document: Done

The Open Group Conference
Enterprise Management for eBusiness in the 21st Century
San Diego, California
24-25 January 2000
How can the emerging eBusiness environment be managed?

CDSA Newsletter
Issue#2 of the Common Data Security Architecture (CDSA) Newsletter has been published.

A/C
The new spec is here
Featured at San Diego

SEMINAR
Making eBusiness Secure: The Role of CDSA and PKI
8th December 1999 : 8:30am - 1:30pm
Santa Clara Marriot, Santa Clara, California

Wireless & Mobile Computing
Visit our new portal
(Members enter here)

Figure 5-1 The Open Group's Web site

As long as the Open Group controlled the core development of the X Window System as public domain software, there was no freely available version of X; it was included only as part of expensive UNIX systems from vendors such as IBM and Hewlett-Packard. These versions of X ran on specialized hardware such as the RS/6000 and PA-RISC microprocessors. The XFree86 Project was started as a nonprofit organization dedicated to creating a version of X that could be used on UNIX systems running on Intel-based computers. The XFree86 Project software was welcomed by Linux developers because it provided the foundation for a graphical interface similar to that used by all other UNIX-like operating systems. X was soon incorporated into most major Linux distributions. You can read more about the XFree86 Project at *www.xfree86.org*.

Concepts Related to X

The success of both the Macintosh and Windows graphical environments was made possible by the fact that the developers of these GUIs had access to detailed information about video graphics hardware. To ensure sales of their products, video card manufacturers needed to ensure that their video cards were compatible with Windows or Macintosh software. To this end, manufacturers agreed to give video card specifications to the software developers. The developers then embedded these specifications in Macintosh and Windows software, thus allowing the operating systems to control the video card.

The developers of X took a different approach. Instead of embedded video card specifications in the operating system kernel, they employed a text-based configuration file whose job was to send controlling information to the video card. By adjusting the information in the configuration file, X could theoretically work with any video card without relying on information from video card manufacturers and constant updates to the X software.

In fact, this system worked well, and continues to work, so long as you want only the minimum level of functionality in your graphical display. This minimal level provides 640 × 480 screen resolution with 256 colors. But every modern computer system is capable of displaying 1024 × 768 screen resolution, often with millions of colors. Hence, the developers of X had to continue developing software to take advantage of the advanced features of newer graphics cards. This process continues to this day. Few standards exist among video card manufacturers, who are all hard at work creating faster, more powerful video cards.

Most of these video card manufacturers were not very forthcoming in providing specifications to the developers at the XFree86 Project, because the largest market for video cards was for Windows-based PCs. This is changing, however. As manufacturers begin to see Linux as a large and growing market, they are more willing to provide information to XFree86 developers. As a result, XFree86 is in a position to create better software for newer video cards, and to do it soon after the video cards are released. Not all vendors provide this information, of course, but the lack of video card specifications is becoming a problem of the past for XFree86 developers.

Components of the X Window System

The original developers of X were very insightful in the way they designed the system, and that early insight has allowed X to continue as a viable technology 15 years after it was first released. This design separates the control of the video card (the computer hardware) from the display of information on the screen. The following components show the further modularization of the parts of the X Window System design, which are incorporated into modern Linux distributions:

- **X server:** this is the program that communicates with the video card to create images on the screen. In software produced by the XFree86 Project, separate X server software packages are used for different types of video cards. For example, one software package, called XFree86_S3, is the X server on computers that have an S3 video chip on their video cards. Several default X servers can be used for less specialized graphics modes. For example, the monochrome, VGA, and SVGA graphics modes are standard on all graphics cards and are managed by X servers that do not rely on a chip-specific X server program. The X server alone does not provide any display on a monitor. A window manager (described later in this list) controls the X server.
- **X client:** this term refers to any kind of graphical application. Essentially, a graphical application runs in the background on a Linux computer and does not in itself have a direct effect on the screen display. Instead, as the application runs, it sends requests to the X server regarding keyboard and mouse input; the X server collects these messages and translates them into images on the screen. The windows and dialog boxes associated with a graphical application are the output of the X server, which is in turn controlled by the X client (the actual application that you are running). The important point here is that X clients do not communicate directly with either your graphics card or your monitor. Although you might find this terminology confusing at first, you should become familiar with it so that you can

use the remote features of X, as described later in this section. An X client is also referred to as an X application.

- **Window manager:** this component is a special-purpose graphical application (that is, a specialized X client) that controls the position and manipulation of the windows within a graphical user interface. Other graphical applications (such as system administration utilities or word processors) rely on the functionality provided by a window manager to direct how the X server displays dialog boxes and performs other standard GUI operations. The window manager gives the X server the ability to draw windows and track mouse movement so that every graphical application does not have to reimplement these functions. At the same time, the window manager provides a library of functions that programmers can use when developing graphical applications.
- **Graphical libraries:** these are collections of programming functions that an X client application can use to more efficiently manage the elements of a graphical environment.
- **Desktop environment:** also known as a desktop interface, this graphical application provides a comprehensive interface, including system menus, desktop icons, and the ability to easily manage files and launch applications. Gnome and KDE are two examples of desktop environments.

Figure 5-2 shows how these components are arranged to interact with each other. The next section explains how these components are actually used in Linux.

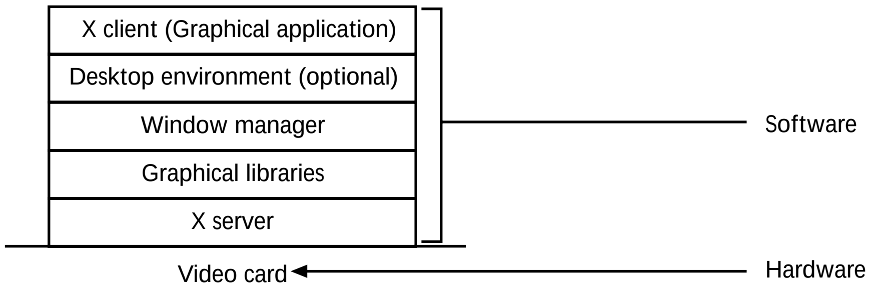


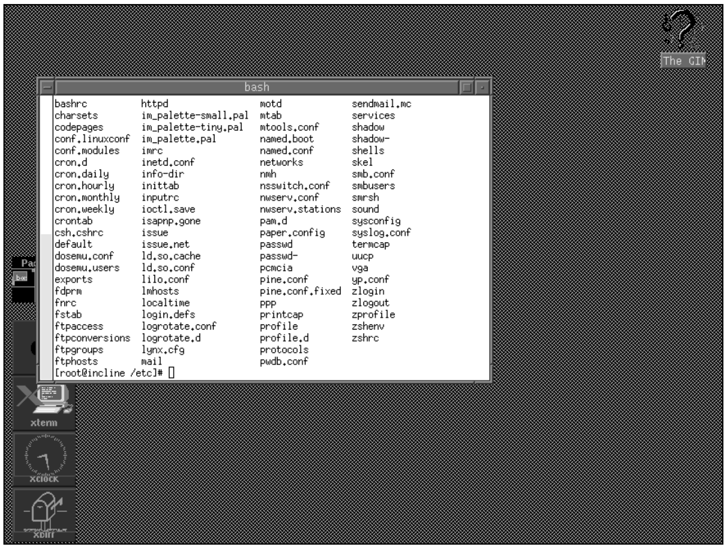
Figure 5-2 Components of the X Window System

How X Components Interact

When you install Linux, the installation program selects which XFree86 server to install based on the computer's video card. The XFree86_SVGA server may be installed, or the XFree86_S3 server, or another server. You can use the `rpm` command to install several X servers if you want to experiment with different servers to see which works best with your video card. Only one X server will be used at a time (as you will see later in the discussion of configuration). Commercial servers from companies such as Xi Graphics (*www.xig.com*) and MetroLink (*www.metrolink.com*) combine X server functionality for all video cards into a single X server program, so you don't need to select among servers when using these products. Future releases of XFree86 will also use this single-server model.

Unlike with Macintosh and Microsoft Windows systems, each user of the X Window System is free to select the interface type he or she prefers. This flexibility is made possible by the numerous window manager programs included with X, each providing a different set of functionality and different look and feel to the graphical environment, and requiring different levels of system resources and expertise to use.

One popular Linux window manager, `fvwm`, is shown in Figure 5-3. When only the window manager is running, you see a blank screen with a color pattern and a mouse pointer. (When you actually open a window manager, you'll see that it alone doesn't provide much of a display on screen, because the window manager is designed mostly to provide services to other programs.) Pressing a mouse button opens a menu from which you can start programs or other functions that control the graphical environment.



```
bashrc          httpd                  motd              sendmail.mc
charsets        im_palette-small.pal   mtab              services
codepages       im_palette-tiny.pal    mtools.conf       shadow
conf.linuxconf  im_palette.pal         named.boot        shadow-
conf.modules    imrc                   named.conf        shells
cron.d          inetd.conf             networks          skel
cron.daily      info-dir               nmh               smb.conf
cron.hourly     inittab                nsswitch.conf     smbusers
cron.monthly    inputrc                nwserv.conf       smrsh
cron.weekly     ioctl.save             nwserv.stations   sound
crontab         isapnp.gone            pam.d             sysconfig
csh.cshrc       issue                  paper.config      syslog.conf
default         issue.net              passwd            termcap
dosemu.conf     ld.so.cache            passwd-           uucp
dosemu.users    ld.so.conf             pcmcia            vga
exports         lilo.conf              pine.conf         yp.conf
fdprm           lmhosts                pine.conf.fixed   zlogin
fnrc            localtime              ppp               zlogout
fstab           login.defs             printcap          zprofile
ftpaccess       logrotate.conf         profile           zshenv
ftpconversions  logrotate.d            profile.d         zshrc
ftpgroups       lynx.cfg               protocols
ftphosts        mail                   pwdb.conf
[root@incline /etc]# 
```

Figure 5-3 The `fvwm` window manager

A few of the window managers available for Linux are described in the following list.

- `twm` (Tab Window Manager): this is a classic UNIX window manager that has been used for many years.
- `fvwm` (Feeble Virtual Window Manager): developed by Robert Nation, this program was the most common window manager for Linux until desktop environments became popular. It includes many of the same characteristics of `twm`, but it requires only about half the memory required to run `twm`. The latest version of `fvwm` is called `fvwm2`. A special version of `fvwm` known as `fvwm95` is designed to emulate the Windows 95 desktop, though tools to emulate the Windows Explorer and other Windows features are not included with `fvwm95`. You can find more information about `fvwm` at *www.fvwm.org*.

- `AMIwm` (Amiga Window Manager): this window manager emulates the look and feel of the old Amiga computers of the late 1980s, specifically the Amiga Workbench.
- `wm2`: a minimal window manager requiring little memory and allowing little configuration.
- `Window Maker` and `AfterStep`: both of these window managers simulate the interface of the NeXT computer.
- `mwm` (Motif Window Manager): a commercial window manager commonly included with commercial UNIX workstations.
- `Enlightenment`: this window manager, sometimes called simply `E`, was based on `fwvm2`. It is used primarily as the window manager for the Gnome Desktop (which is discussed later in this chapter).
- `olwm` (OpenLook Window Manager): the OpenLook interface style was created by Sun Microsystems and is still used primarily by developers wishing to emulate the look and feel of a Sun UNIX workstation.

Why would you choose one window manager over another? The choice is a matter of personal preference based on the appearance of the window manager and the features that it offers. A hallmark of Linux is that it allows the flexibility to configure components (such as the window manager) according to your personal preferences.

Many different graphical libraries are also available for X. For the most part, each of these libraries was originally created to simplify the development of a specific application. The particularly useful libraries were used by developers of other applications. A graphical library consists of a few files containing programming commands that graphical applications can access. A graphical library is installed on a Linux system just like any other application (using an `rpm` command, for example). But a graphical library is not launched directly—it only provides tools for other applications. The two graphical libraries that are most widely known for Linux at this time are listed here:

- The Qt library was developed by a company in Norway called Troll Tech. Qt was used as the foundation of the KDE Desktop, which is the most popular graphical environment used on Linux. (See *www.kde.org* for more information.) KDE is described in more detail later in this chapter.
- The GTK+ library was developed by Spencer Kimball and Peter Mattis at the University of California at Berkeley as the foundation of the Gimp graphics application. (Gimp is a program similar to Adobe Photoshop.) Subsequently, GTK+ was used to create the Gnome Desktop, which is the second of the two major graphical environments used on Linux systems today. (See *www.gnome.org* for more information.) Gnome is described in more detail later in this chapter.

Using X Remotely

Although X supports many features (described in the configuration sections that follow), the remote capabilities of X deserve special mention. With X, you can display graphical applications remotely; that is, you can launch an application (an X client) on one computer, but specify that the application should be displayed on a different computer. For example, suppose you have WordPerfect for Linux installed on one computer (call it computer A). Many different users can log in to computer A and start a copy of WordPerfect at the same time. All of the copies of WordPerfect are running on computer A, but the WordPerfect program appears on computers B, C, D, and so forth, where all of the users are physically located.

Each computer (B, C, D in this example) must be running an X server program. The X server receives requests from the X client (WordPerfect) to display windows and collect keyboard and mouse input. Figure 5-4 illustrates the arrangement of the X server and X client.

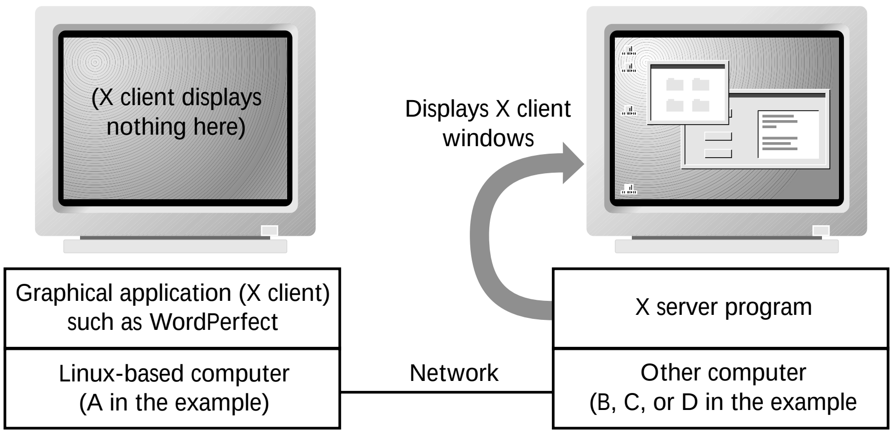
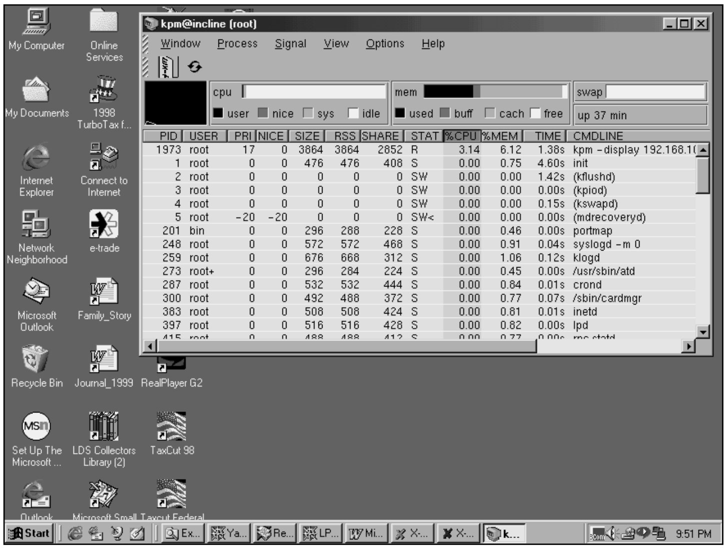

(X client displays nothing here)
Displays X client windows
Graphical application (X client) such as WordPerfect
Linux-based computer (A in the example)
Network
X server program
Other computer (B, C, or D in the example

Figure 5-4 An X client running on a remote X server

The X server used to provide the screen display for an X client can be located on any type of computer. For example, X server programs are available for Microsoft Windows computers. By installing one of these programs, you can launch an application on a Linux computer system and have that application displayed on a Windows computer, as shown in Figure 5-5. You can learn about two companies that sell X servers for Windows by visiting *www.starnet.com* (look for the X-Win product) and *www.hummingbird.com* (look for the Exceed product).



```
  PID  USER   PRI  NICE  SIZE   RSS SHARE STAT  %CPU %MEM   TIME  CMDLINE
 1973  root    17     0  3864  3864  2852 R     3.14  6.12  1.38s kpm -display 192.168.1(
    1  root     0     0   476   476   408 S     0.00  0.75  4.60s init
    2  root     0     0     0     0     0 SW    0.00  0.00  1.42s (kflushd)
    3  root     0     0     0     0     0 SW    0.00  0.00  0.00s (kpiod)
    4  root     0     0     0     0     0 SW    0.00  0.00  0.15s (kswapd)
    5  root   -20   -20     0     0     0 SW<   0.00  0.00  0.00s (mdrecoveryd)
  201  bin      0     0   296   288   228 S     0.00  0.46  0.00s portmap
  248  root     0     0   572   572   468 S     0.00  0.91  0.04s syslogd -m 0
  259  root     0     0   676   668   312 S     0.00  1.06  0.12s klogd
  273  root+    0     0   296   284   224 S     0.00  0.45  0.00s /usr/sbin/atd
  287  root     0     0   532   532   444 S     0.00  0.84  0.01s crond
  300  root     0     0   492   488   372 S     0.00  0.77  0.07s /sbin/cardmgr
  383  root     0     0   508   508   424 S     0.00  0.81  0.01s inetd
  397  root     0     0   516   516   428 S     0.00  0.82  0.00s lpd
```

Figure 5-5 An X server displaying a Linux program within Windows

RUNNING THE X WINDOW SYSTEM

Though the flexibility and features of X are impressive, they come at a price: the X Window System can be difficult to configure. The following list explains some of the issues that make setting up X such a challenge:

- Video card manufacturers do not focus on creating X software for their hardware devices, so you must make adjustments to fit the video hardware's requirements.
- The various components of X (such as window managers and graphical libraries) are interchangeable, developed by separate groups, and based on completely different configuration models, requiring that you learn multiple versions of each component in order to support a variety of user preferences and application needs.
- X supports a multiuser, networked environment, which means you may need to set up and maintain numerous configuration files. (These configuration files are described later in this chapter.)

You learned the basics of configuring X in Chapter 3. The following sections provide additional guidance on getting X installed and working.

Installing and Configuring X

When you installed Linux, at least one X server from the XFree86 set of X servers was probably installed, along with the supporting packages needed to run graphical applications. You can review the packages that are installed on your system by using the command `rpm -qa | grep XFree`. The output of this command for a typical Red Hat Linux installation is shown here:

```
XFree86-75dpi-fonts-3.3.5-3
XFree86-SVGA-3.3.5-3
XFree86-libs-3.3.5-3
XFree86-xfs-3.3.5-3
XFree86-3.3.5-3
XFree86-devel-3.3.5-3
```

If a list similar to this does not appear on your system, you should use the `rpm` command with the `-Uvh` options (as described in Chapter 4) to install these packages from the Linux CD (or other installation source files available to you). If you cannot successfully configure X following the steps in this section, you should consider trying to upgrade the XFree86 server to a later version to add support for the video card you are configuring. Visit the Web site *www.xfree86.org* to learn the version number of the latest release of the software. If the output of the `rpm` command shows an older version of XFree86, you may need to upgrade to the latest version. (In the sample `rpm` output above, the version number of XFree86 is 3.3.5.) You can do this by downloading software from the *ftp.xfree86.org* Internet site or, in most cases, by visiting the Web site of your Linux vendor (such as *ftp.redhat.com*) and obtaining updated software packages for the XFree86 software. In some cases, you will not need a completely new set of XFree86 packages; you may simply be able to add a new X server program to support the video card. Again, you can locate new X servers on the XFree86 Web site or on the Web site of your Linux vendor.

TIP

You can learn about configuration options for X by reviewing the X online manual page (enter the command `man X`). Additional manual pages are available for each X server (for example, enter the command `man XFree86_SVGA`).

The X software is normally located in the directory `/usr/X11R6`. This directory is sometimes called the X-root directory. The filename `X11R6` refers to the fact that the features of X used in current X servers are based upon X version 11, release 6. You will hear people who work with many types of Linux and UNIX systems refer to "ex-eleven-are-five" or "ex-eleven-are-six" when describing their version of X. The developers of X have stopped advancing the version number (it stays at 11), but the release number does change occasionally (every few years).

In addition to the XFree86 software, you need at least a window manager program, and perhaps other software packages, depending on what software you want to run on the system. All general-purpose Linux distributions install these components by default. For example, you can enter the command `rpm -q fvwm` to see if the `fvwm` window manager is installed. Using the command `rpm -qa | grep kde` will list all of the packages related to the KDE Desktop. You can use similar commands to check for the Gnome Desktop or for any graphical libraries or window managers.

TIP Some specialized Linux distributions, such as the Linux Router Project (see *www.linuxrouter.org*), do not include X. On any Linux system, you can choose not to run X to save system resources (both memory and CPU processing time).

Chapter 3 mentioned several utilities you can use to set up X. The configuration described in Chapter 3 and in this section is only for configuring the X server to access the video card. After the X server is configured, the window manager or desktop environment that you use will work without additional configuration. You can also make changes in the configuration of a window manager or desktop environment. These changes are relatively simple and straightforward because they do not require that you know any additional information about the computer's hardware. But before these other components can work, you must first properly configure the X server.

You normally configure the X Window System when you install Linux. But as this is sometimes not possible because of problems identifying the video card or configuring its use, you may need to set up the X configuration file after installing the rest of the system.

It's sometimes useful to launch X several times with different configurations until you find one that works. If the display is garbled or the screen simply goes blank when you launch X (using the commands given later in this chapter), try pressing the key combination Ctrl+Alt+Backspace to exit X and return to character mode. If the keyboard does not respond but the computer is connected to a network, you should be able to log in to the system from another computer and use the `kill` command to end the X server program. (See Chapter 10 for details on using the `kill` command.)

The Configuration File

The configuration file for XFree86 is located either in the `/etc` directory, in the `/etc/X11` directory, or in the `/usr/X11R6/lib/X11` directory. The configuration file is called `XF86Config`. (Note the use of upper- and lowercase.) A sample configuration file is shown here:

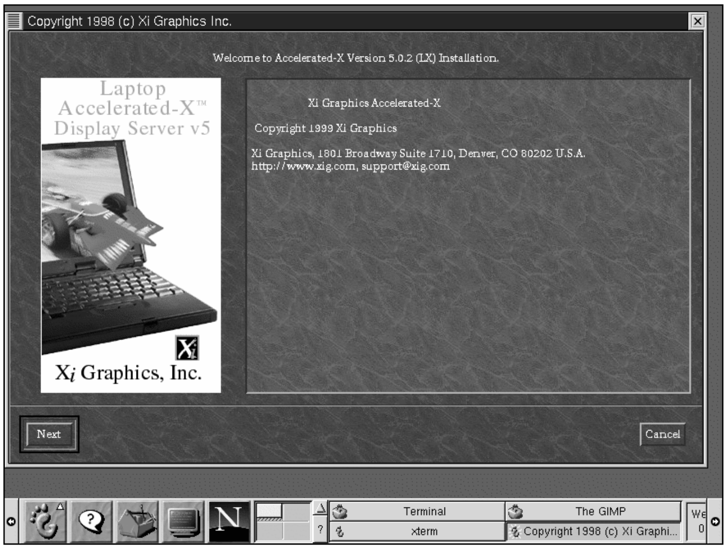
```
# See 'man XF86Config' for info on the format of this file
#

Section "Files"
    RgbPath "/usr/X11R6/lib/X11/rgb"
    FontPath "/usr/X11R6/lib/X11/fonts/Type1"
    FontPath "/usr/X11R6/lib/X11/fonts/Speedo"
    FontPath "/usr/X11R6/lib/X11/fonts/misc"
    FontPath "/usr/X11R6/lib/X11/fonts/75dpi"
EndSection


Section "ServerFlags"
EndSection
```

```
Section "Keyboard"
    Protocol "Standard"
    XkbRules "xfree86"
    XkbModel "pc101"
    XkbLayout "us"
EndSection


Section "Pointer"
    Protocol "microsoft"
    Device "/dev/ttyS0"

    Emulate3Buttons
EndSection


Section "Monitor"
    Identifier "Primary Monitor"
    VendorName "Typical monitors"
    ModelName "800x600, 60Hz"
    HorizSync 31.5-37.9
    VertRefresh 55-90
    Modeline "800x600/60Hz" 40 800  840  968 1056
         600  601  605  628
+HSync +VSync
EndSection


Section "Device"
    Identifier "Primary Card"
    VendorName "Unknown"
    BoardName "None"
    VideoRam 2048
EndSection


Section "Screen"
    Driver "VGA16"
    Device "Primary Card"
    Monitor "Primary Monitor"
    SubSection "Display"
        Depth 4
        Modes "640x480/60Hz"
    EndSubSection
EndSection


Section "Screen"
    Driver "SVGA"
    Device "Primary Card"
    Monitor "Primary Monitor"
    DefaultColorDepth 16
    SubSection "Display"
        Depth 8
        Modes "800x600/60Hz"
              Virtual 800 600
    EndSubSection
    SubSection "Display"
        Depth 15
        Modes "800x600/60Hz"
              Virtual 800 600
    EndSubSection
    SubSection "Display"
        Depth 16
        Modes "800x600/60Hz"
              Virtual 800 600
    EndSubSection
    SubSection "Display"
        Depth 24
        Modes "800x600/60Hz"
              Virtual 800 600
    EndSubSection
    SubSection "Display"
        Depth 32
        Modes "800x600/60Hz"
              Virtual 800 600
    EndSubSection
EndSection

Section "Screen"
    Driver "accel"
    Device "Primary Card"
    Monitor "Primary Monitor"
    DefaultColorDepth 16
    SubSection "Display"
        Depth 8
        Modes "800x600/60Hz"
              Virtual 800 600
    EndSubSection
    SubSection "Display"
        Depth 15
        Modes "800x600/60Hz"
              Virtual 800 600
    EndSubSection
    SubSection "Display"
        Depth 16
        Modes "800x600/60Hz"
              Virtual 800 600
    EndSubSection
    SubSection "Display"
        Depth 24
        Modes "800x600/60Hz"
              Virtual 800 600
    EndSubSection
    SubSection "Display"
        Depth 32
        Modes "800x600/60Hz"
              Virtual 800 600
    EndSubSection
EndSection

# End of automatically generated file
```

Notice that the `XF86Config` file contains specifications for display fonts, instructions for using the keyboard and mouse, details about the monitor and video card capabilities, and information on combining the video card and monitor details to create a specific view (such as 1024 × 768 resolution with 256 colors).

The sample file shown above does not include comments, which you will find in many `XF86Config` files. These comments can help you understand the different sections of the configuration file, and they provide examples of alternative configurations.

Commercial X servers do not rely on the `XF86Config` file. Each uses a different configuration file with a different format. For example, the Accelerated-X server from Xi Graphics uses the configuration file `/etc/Xaccel.ini`. To make the task of configuring them easier, commercial X servers include configuration tools similar to the XFree86 configuration tools described later in this chapter. One such tool, which provides configuration options for Accelerated-X, is shown in Figure 5-6. In general, when working with a commercial X server, you should review documentation accompanying the product to learn how to use the product's configuration utility before preparing a configuration file.


Copyright 1998 (c) Xi Graphics Inc.

Welcome to Accelerated-X Version 5.0.2 (LX) Installation.

Xi Graphics Accelerated-X

Copyright 1999 Xi Graphics

Xi Graphics, 1801 Broadway Suite 1710, Denver, CO 80202 U.S.A.
http://www.xig.com, support@xig.com

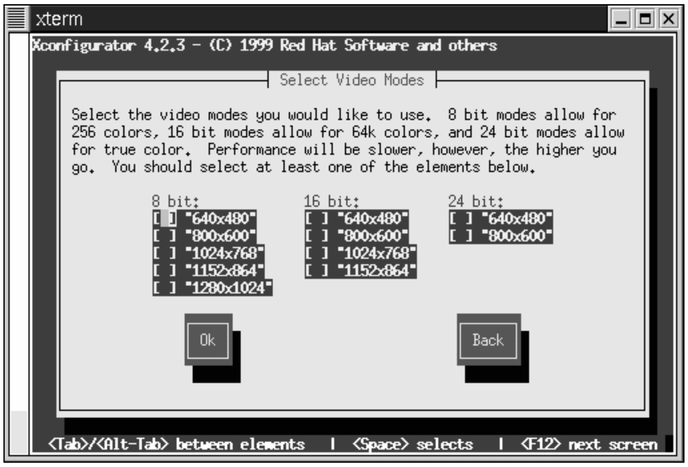
Figure 5-6 The configuration utility for Accelerated-X

Using `Xconfigurator`

The **`Xconfigurator`** program is a utility for creating an `XF86Config` file for your XFree86 X server. It was developed by Red Hat Software and is executed automatically as part of the installation of Red Hat Linux. You can also run this program after completing the installation in order to complete or redo the X configuration. To start the program, enter the program's name on a command line (note the capital `X`):

```
Xconfigurator
```

The interface to `Xconfigurator` will be familiar to you if you have installed Red Hat Linux. `Xconfigurator` lets you select a video card from a list and then gives you the option of manually setting additional options or having `Xconfigurator` probe the video card to determine the best settings. Figure 5-7 shows a sample screen from `Xconfigurator`, where you can select the screen resolutions to use with different color depths. Detailed steps for using `Xconfigurator` are provided as part of the installation instructions in Chapter 3.



```
xterm
Xconfigurator 4.2.3 - (C) 1999 Red Hat Software and others

                     Select Video Modes

   Select the video modes you would like to use.  8 bit modes allow for
   256 colors, 16 bit modes allow for 64k colors, and 24 bit modes allow
   for true color.  Performance will be slower, however, the higher you
   go.  You should select at least one of the elements below.

          8 bit:              16 bit:             24 bit:
          [ ] "640x480"       [ ] "640x480"       [ ] "640x480"
          [ ] "800x600"       [ ] "800x600"       [ ] "800x600"
          [ ] "1024x768"      [ ] "1024x768"
          [ ] "1152x864"      [ ] "1152x864"
          [ ] "1280x1024"

              Ok                                   Back

 <Tab>/<Alt-Tab> between elements   |  <Space> selects   |  <F12> next screen
```

Figure 5-7 Selecting screen resolutions in Xconfigurator

Although `Xconfigurator` is a helpful tool, it is not always the best way to set up X, because it may not provide all the options you need. As a result, you may find that after choosing the video card in your system and selecting the options that seem correct, X still will not function. (See the section "Launching X" later in this chapter.) The problem with `Xconfigurator` is that it does not provide sufficient configuration detail to let you select the options that would result in a correctly functioning X server; neither does it have the ability to set up the video card without asking you numerous questions. If `Xconfigurator` does not successfully set up X, try using one of the other tools described in the following sections.

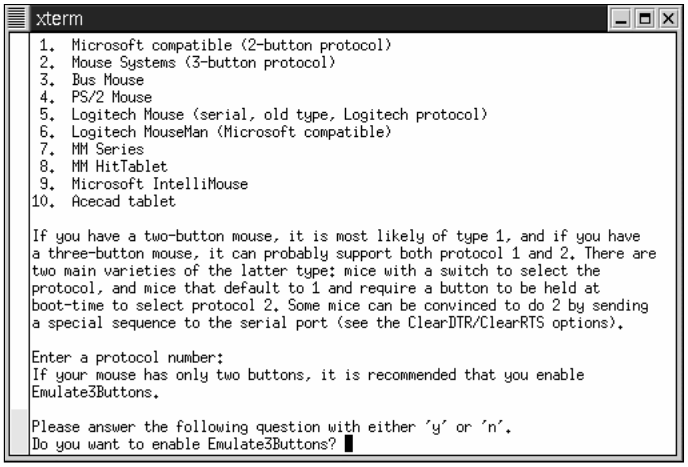
Using `xf86config`

The XFree86 software comes with a command-line configuration utility called **`xf86config`** (all lowercase). After launching this utility you must read detailed information presented on screen for each option and then answer questions as you are prompted. The `xf86config` utility provides complete access to the features of the XFree86 software. If you take the time to read the information presented on screen and you know the hardware details of your video card (such as the video chipset and the amount of video RAM), `xf86config` is very likely to correctly configure any video card supported by an XFree86 X server.

The `xf86config` utility is included on a standard Red Hat Linux system. Many other Linux distributions also include this utility. To launch `xf86config`, enter the utility's name at any Linux command line (note that it is all lowercase):

```
xf86config
```

Figure 5-8 shows one of the questions asked by `xf86config`, along with with a list of options to choose from and detailed instructions about how to select the correct option.



```
 1.  Microsoft compatible (2-button protocol)
 2.  Mouse Systems (3-button protocol)
 3.  Bus Mouse
 4.  PS/2 Mouse
 5.  Logitech Mouse (serial, old type, Logitech protocol)
 6.  Logitech MouseMan (Microsoft compatible)
 7.  MM Series
 8.  MM HitTablet
 9.  Microsoft IntelliMouse
10.  Acecad tablet

If you have a two-button mouse, it is most likely of type 1, and if you have
a three-button mouse, it can probably support both protocol 1 and 2. There are
two main varieties of the latter type: mice with a switch to select the
protocol, and mice that default to 1 and require a button to be held at
boot-time to select protocol 2. Some mice can be convinced to do 2 by sending
a special sequence to the serial port (see the ClearDTR/ClearRTS options).

Enter a protocol number:
If your mouse has only two buttons, it is recommended that you enable
Emulate3Buttons.

Please answer the following question with either 'y' or 'n'.
Do you want to enable Emulate3Buttons? █
```

Figure 5-8 The `xf86config` utility

The following list summarizes the information that `xf86config` requests during the configuration process. The exact questions posed and the order of those questions depends on your answers (which in turn depend on which video card you are configuring).

- Mouse protocol
- Mouse device
- Keyboard settings
- Monitor scan frequencies
- Video card description

- XFree86 server to use
- Amount of video memory
- The clockchip setting
- Resolution and number of colors to use on screen

Using `XF86Setup`

Although the `xf86config` utility is an effective means of configuring X, it requires you to read a lot of information and answer many questions that may not be relevant to your video configuration. As an alternative, XFree86 also includes a graphical configuration utility called **`XF86Setup`**. This utility is not included with Red Hat Linux 6.0, but it is found on many Linux distributions and is available through the XFree86 Web site. `XF86Setup` uses the VGA graphics mode, which is a low-resolution mode supported by virtually all video cards. This mode provides a screen with 640 × 480 resolution on which you can use a mouse to select items to configure the X server for higher resolutions.

To start `XF86Setup`, enter the utility's name on any Linux command line. A series of text-mode messages prompt you for confirmation before switching to the VGA mode in which the configuration options are displayed. The first screen, used for configuring the mouse, requires you to use the keyboard (the Tab, Spacebar, and Enter keys) to select the correct mouse configuration. This configuration screen is shown in Figure 5-9. After you have configured the mouse and chosen the `Apply` button to activate it, you can use the mouse to select other configuration options.

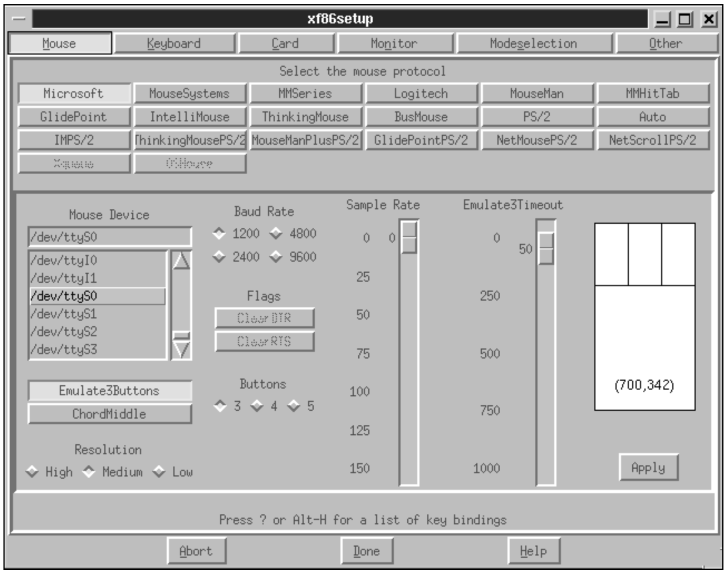
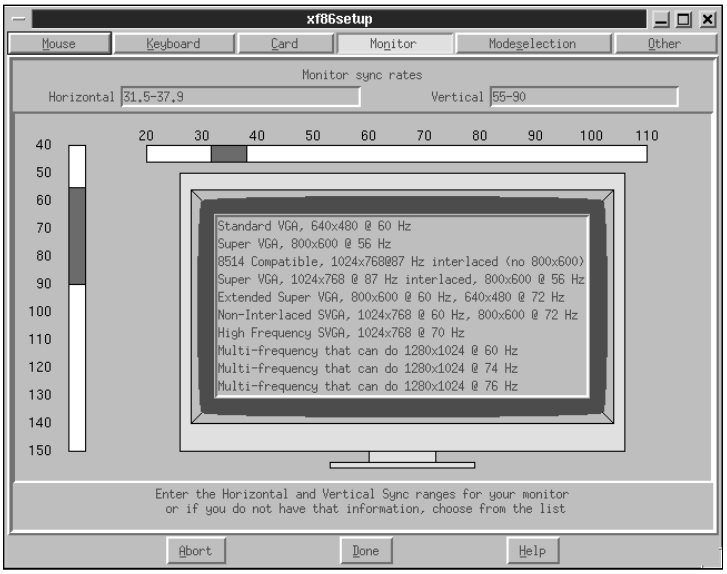


Figure 5-9 The `Mouse` configuration tab in `XF86Setup`

On the `Monitor` tab of `XF86Setup`, which is shown in Figure 5-10, you select the monitor scan values, as described in Chapter 3. Recall that if you select a higher scan rate or resolution than your monitor supports, you risk damaging your monitor.



Monitor sync rates

Horizontal 31.5-37.9 Vertical 55-90

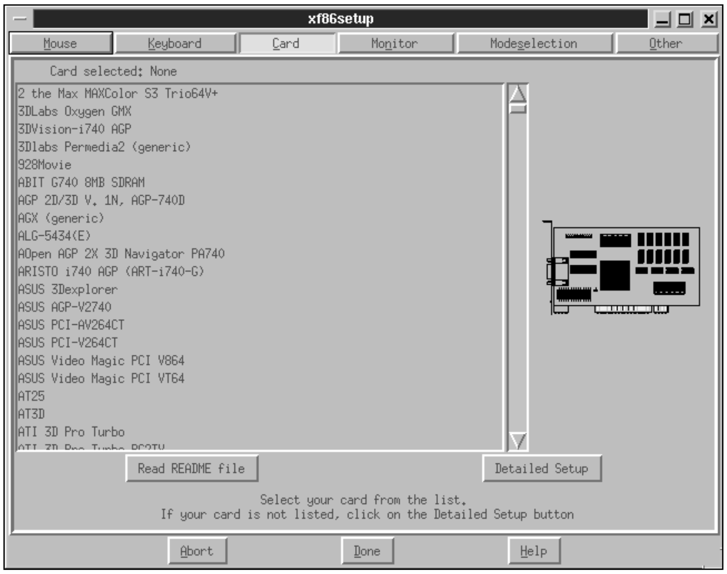
```
Standard VGA, 640x480 @ 60 Hz
Super VGA, 800x600 @ 56 Hz
8514 Compatible, 1024x768@87 Hz interlaced (no 800x600)
Super VGA, 1024x768 @ 87 Hz interlaced, 800x600 @ 56 Hz
Extended Super VGA, 800x600 @ 60 Hz, 640x480 @ 72 Hz
Non-Interlaced SVGA, 1024x768 @ 60 Hz, 800x600 @ 72 Hz
High Frequency SVGA, 1024x768 @ 70 Hz
Multi-frequency that can do 1280x1024 @ 60 Hz
Multi-frequency that can do 1280x1024 @ 74 Hz
Multi-frequency that can do 1280x1024 @ 76 Hz
```

Enter the Horizontal and Vertical Sync ranges for your monitor
or if you do not have that information, choose from the list

Figure 5-10 The `Monitor` tab in `XF86Setup`

The most challenging part of using `XF86Setup` is selecting the correct settings on the `Card` tab. You can view this tab two different ways: in Card List view, or in Detailed view. You only need to complete one of these views to finish the video card configuration.

In Card List view, shown in Figure 5-11, you see a long list of video cards arranged by manufacturer and model. (If necessary, you can choose the `Card List` button to switch to this view.) If the video card you are configuring is included in this list, simply select it. If the video card is not listed, choose the `Detailed Setup` button to switch to the `Card` tab's Detailed view, which is shown in Figure 5-12. In Detailed view, you must define the characteristics of the video card by selecting options such as the amount of video memory. Fortunately, many of these fields include the `Probed` option; selecting this option tells the `XF86Setup` program to attempt to determine the correct value for you. If the program is not able to do this correctly, however, you may need to enter an explicit value to get the X server to function.

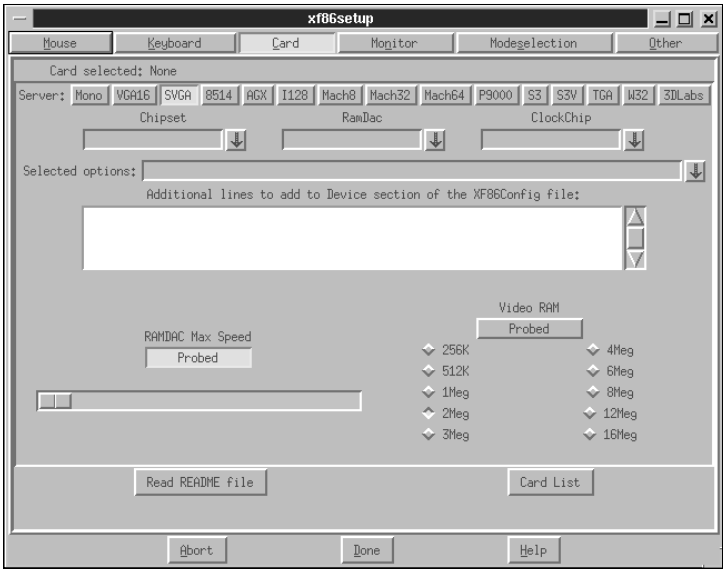


Card selected: None

2 the Max MAXColor S3 Trio64V+
3DLabs Oxygen GMX
3DVision-i740 AGP
3Dlabs Permedia2 (generic)
928Movie
ABIT G740 8MB SDRAM
AGP 2D/3D V. 1N, AGP-740D
AGX (generic)
ALG-5434(E)
AOpen AGP 2X 3D Navigator PA740
ARISTO i740 AGP (ART-i740-G)
ASUS 3Dexplorer
ASUS AGP-V2740
ASUS PCI-AV264CT
ASUS PCI-V264CT
ASUS Video Magic PCI V864
ASUS Video Magic PCI VT64
AT25
AT3D
ATI 3D Pro Turbo

Select your card from the list.
If your card is not listed, click on the Detailed Setup button

Figure 5-11 The Card List view of the `Card` tab



Card selected: None

Server: Mono VGA16 SVGA 8514 AGX I128 Mach8 Mach32 Mach64 P9000 S3 S3V TGA W32 3DLabs

Chipset
RamDac
ClockChip

Selected options:

Additional lines to add to Device section of the XF86Config file:

RAMDAC Max Speed: Probed

Video RAM: Probed

256K
512K
1Meg
2Meg
3Meg
4Meg
6Meg
8Meg
12Meg
16Meg

Figure 5-12 The Detailed view of the `Card` tab

To complete the configuration using `XF86Setup`, choose the `Mode` tab and select a screen resolution. The most commonly used screen resolution is 1024 × 768. Unless you have special instructions from another source (such as your video card manufacturer), you don't need to set any of the options on the `Other` tab, though you can explore them if you wish.

When you choose the `Done` button and confirm that you are ready to exit `XF86Setup`, the utility creates a temporary `XF86Config` file for you and starts the X server to test the configuration. If the server starts successfully, you see the `xvidtune` program's graphical screen. You can use the options in this program to adjust the position of the picture on your monitor, but this is usually not necessary. (You can generally use the controls located on the front of most monitors to control the position of the screen.) Figure 5-13 shows a view of the `xvidtune` program. When you have finished exploring the options in `xvidtune`, choose `Save and Exit` to exit the X server and save the new `XF86Config` file.

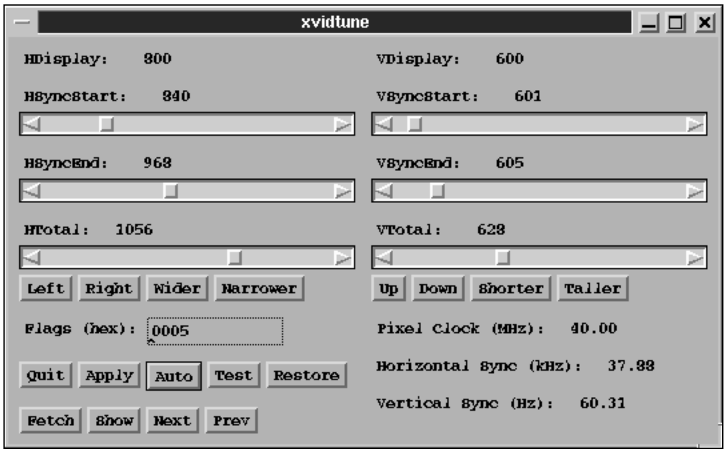

```
HDisplay:   800          VDisplay:   600
HSyncStart:   840        VSyncStart:   601
HSyncEnd:   968          VSyncEnd:   605
HTotal:  1056            VTotal:   628
Flags (hex): 0005        Pixel Clock (MHz):   40.00
                         Horizontal Sync (kHz):   37.88
                         Vertical Sync (Hz):   60.31
```

Figure 5-13 The `xvidtune` program

Occasionally the `XF86Setup` program appears to work correctly (that is, the test screen at the end of the process displays correctly), but then when you launch X, as described later in this chapter, the window manager screen or desktop environment does not appear as it should. In this case you can exit X and restart the `XF86Setup` program to try a different set of configuration options.

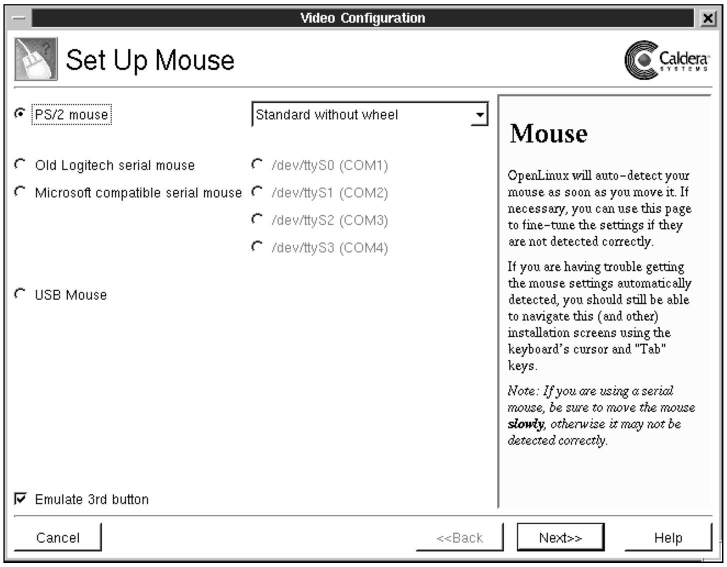
Using `lizardx`

The easiest method of configuring X is to use the **`lizardx`** program from Caldera Systems. This program, like the `XF86Setup` program, provides a graphical interface in which you select a few simple options. But `lizardx` (short *for Linux installation wizard for X*) doesn't require you to know anything about your video card. Its autoprobing features are the best available for Linux.

`Lizardx` was taken from the installation program for Caldera System's OpenLinux product. But `lizardx` is a separate program that you can execute from a command line in any Linux distribution (after you have downloaded the free `lizardx` program from the Caldera Systems FTP site at *ftp.calderasystems.com*). Start the program using this command:

```
lizardx
```

When the first screen of `lizardx` appears (as shown in Figure 5-14), move the mouse around slowly to autoconfigure it. Then choose the `Next` button to proceed to another screen where you can select a keyboard model and layout. Most users can accept the default settings and simply choose `Next` to continue.



(● PS/2 mouse) Standard without wheel
○ Old Logitech serial mouse
○ Microsoft compatible serial mouse
○ /dev/ttyS0 (COM1)
○ /dev/ttyS1 (COM2)
○ /dev/ttyS2 (COM3)
○ /dev/ttyS3 (COM4)
○ USB Mouse
☑ Emulate 3rd button

Mouse

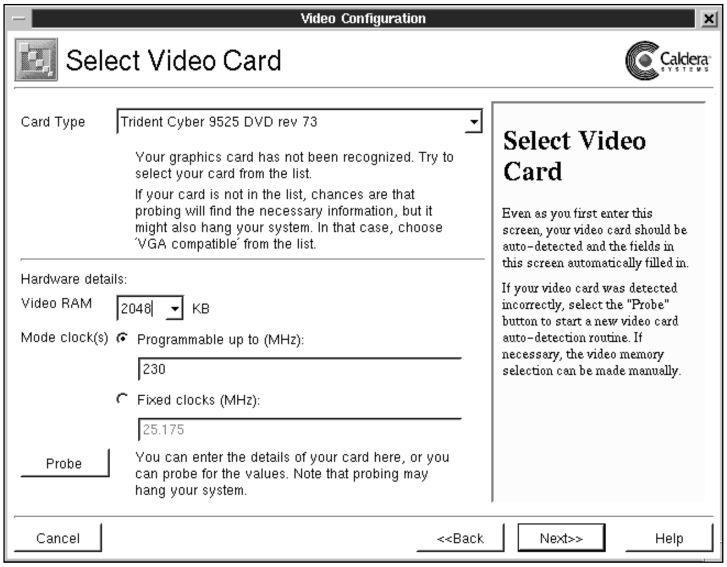
OpenLinux will auto-detect your mouse as soon as you move it. If necessary, you can use this page to fine-tune the settings if they are not detected correctly.

If you are having trouble getting the mouse settings automatically detected, you should still be able to navigate this (and other) installation screens using the keyboard's cursor and "Tab" keys.

Note: If you are using a serial mouse, be sure to move the mouse ***slowly****, otherwise it may not be detected correctly.*

Figure 5-14 The `lizardx` mouse configuration screen

The next screen is where the wizardry of `lizardx` is evident. In this `Select Video Card` screen, you'll find that the video card has already been autodetected and appears in the drop-down list at the top of the screen (see Figure 5-15). To finish the configuration of the video card, choose the `Probe` button; then confirm that the probing can begin by choosing `OK` in the message box that appears. The screen goes black, then returns after a moment with the message `Probing is complete and apparently successful`. Choose the `Next` button to continue.



Your graphics card has not been recognized. Try to select your card from the list.

If your card is not in the list, chances are that probing will find the necessary information, but it might also hang your system. In that case, choose 'VGA compatible' from the list.

Hardware details:

Video RAM: 2048 KB

Mode clock(s): [✓ Programmable up to (MHz):] 230

Fixed clocks (MHz): 25.175

You can enter the details of your card here, or you can probe for the values. Note that probing may hang your system.

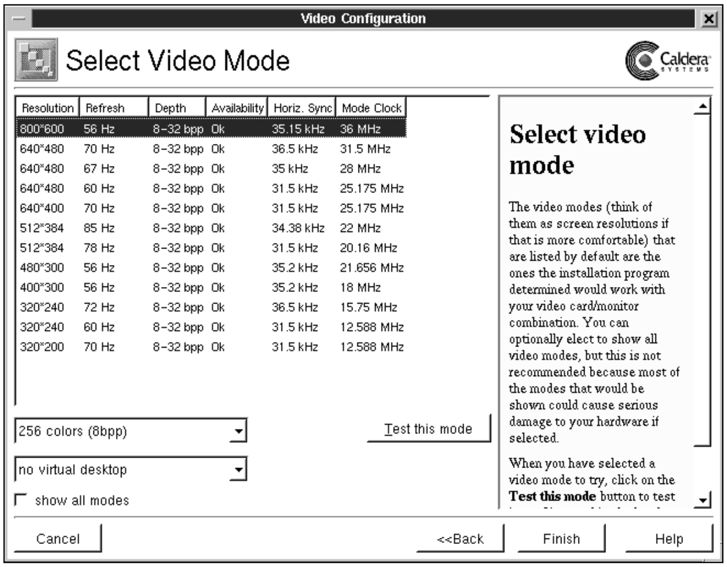
Select Video Card

Even as you first enter this screen, your video card should be auto-detected and the fields in this screen automatically filled in.

If your video card was detected incorrectly, select the "Probe" button to start a new video card auto-detection routine. If necessary, the video memory selection can be made manually.

Figure 5-15 The `Select Video Card` screen in `lizardx`

In the next configuration screen you select a monitor model from a list of thousands of models, all arranged by manufacturer. A short list of standard or default monitor types is included at the top of the list in case the monitor you need is not listed. When you select `Next`, the `Select Video Mode` screen appears, as shown in Figure 5-16. On this screen, `lizardx` presents a list of video modes that are available based on your video card and monitor. First, choose the number of colors you want to include by selecting an item in the drop-down list in the lower-left part of the screen. (The number of colors you select may change the list of available screen resolutions, depending on the amount of video memory on the video card.) Then choose one of the screen modes that looks appealing (based on the screen resolution). Finally, choose the `Test` button, and then choose `Yes` in the message box that appears. If you see a functional desktop screen after a few moments, the graphics are configured, and you can choose the `Finish` button to complete the configuration. If you do not see a desktop interface displayed, wait a moment until the `lizardx` screen reappears; then choose another mode in the list and select the `Test` button again. Repeat this process until you find a mode that works well.



Resolution	Refresh	Depth	Availability	Horiz. Sync	Mode Clock
800*600	56 Hz	8–32 bpp	Ok	35.15 kHz	36 MHz
640*480	70 Hz	8–32 bpp	Ok	36.5 kHz	31.5 MHz
640*480	67 Hz	8–32 bpp	Ok	35 kHz	28 MHz
640*480	60 Hz	8–32 bpp	Ok	31.5 kHz	25.175 MHz
640*400	70 Hz	8–32 bpp	Ok	31.5 kHz	25.175 MHz
512*384	85 Hz	8–32 bpp	Ok	34.38 kHz	22 MHz
512*384	78 Hz	8–32 bpp	Ok	31.5 kHz	20.16 MHz
480*300	56 Hz	8–32 bpp	Ok	35.2 kHz	21.656 MHz
400*300	56 Hz	8–32 bpp	Ok	35.2 kHz	18 MHz
320*240	72 Hz	8–32 bpp	Ok	36.5 kHz	15.75 MHz
320*240	60 Hz	8–32 bpp	Ok	31.5 kHz	12.588 MHz
320*200	70 Hz	8–32 bpp	Ok	31.5 kHz	12.588 MHz

256 colors (8bpp)

no virtual desktop

☐ show all modes

Select video mode

The video modes (think of them as screen resolutions if that is more comfortable) that are listed by default are the ones the installation program determined would work with your video card/monitor combination. You can optionally elect to show all video modes, but this is not recommended because most of the modes that would be shown could cause serious damage to your hardware if selected.

When you have selected a video mode to try, click on the **Test this mode** button to test

Figure 5-16 The `Select Video Mode` screen in `lizardx`

Configuring X Using Other Resources

Just as you likely consulted your system administrator or Internet Service Provider when gathering information about your network, you may need to contact knowledgeable people to help you set up a video card to work in Linux. To get help configuring X, try the following:

- Ask another person who has successfully installed Linux several times to help you configure X. Sometimes options that are unfamiliar to you will be well known to a person who has already configured a similar system.
- Visit the XFree86 Web site and search for information about the video card you are trying to configure.
- Visit the Web site of your Linux vendor and search for information about configuring X.
- Use a newsgroup reader to explore messages on the *comp.os.linux* newsgroups and related newsgroups for X. Some of these are dedicated to discussions of the X Window System. Read a few messages; then post a brief, polite message asking for guidance. Include as much detail about the computer system and video card as you have available.
- Buy a commercial X server (after checking that support for the troublesome video card is included). Contact the vendor for support if you have trouble installing the product.
- Visit Web sites dedicated to using Linux. In particular, try the Linux on Laptops page at *www.cs.utexas.edu/users/kharker/linux-laptop/*. This page has links for hundreds of laptop models, where owners of those laptops have posted suggestions

(and often sample `XF86Config` files) for making X work with specific models of laptops.

- Visit the Web site of the computer vendor and search for information in the technical support database. As more vendors add support for Linux, you are more likely to find helpful information from computer system vendors.

If you have easy access to any of these resources, you might choose to consult them before trying the different programs outlined earlier in this chapter.

Launching X

The standard method of starting X is to log in to Linux on a character-mode console and then execute the `startx` command. This command automatically executes a number of other commands that launch the X server and run the programs that make up the graphical environment. In this section, you will learn how all those programs interact.

TIP

Depending on how your Linux distribution is configured, you may see a graphical login screen when the computer starts. The graphical login screen is called the X display manager, or `xdm`. The last section of this chapter describes how to use and configure `xdm`.

The `startx` program is actually a small script located in `/usr/X11R6/bin`. This script launches another program called `xinit`, which looks for several scripts in various locations in the Linux directory structure, including `~/.xinitrc`, `~/.Xclients`, `/etc/X11/xinit/xinitrc`, and `/etc/X11/xinit/Xclients`. A system administrator can place scripts in a user's home directory (`~/.xinitrc` and `~/.Xclients`) to define a unique graphical configuration for that user. All of the scripts listed here specify which programs should start along with X. For the most part, these programs are X clients. (As explained earlier in this chapter, an X client is an application that runs in X.) Each X client is started as a background application. A **background application** is an application that does not prevent the program that started it (the `xinit` program in this case) from going on to other tasks (launching other X clients). In this case, the `xinit` scripts can start an X client and then go on to start another (and another, and another, and so on) without waiting for the first one to finish execution. The X clients that `xinit` starts from the initialization scripts include programs such as a file manager, an on-screen clock, and a toolbar. These X clients provide a minimal set of graphical tools as the window manager is started.

The last X client started is the window manager, which is responsible for controlling the graphical screen. The window manager features are used by all of the X applications that are started before the window manager; the other X applications wait until the window manager is started before trying to display information on screen.

A very simple script that `xinit` uses to launch one X client and a window manager might look like this:

```
xterm &
fvwm
```

If this script were used by `xinit` to start X, a single `xterm` (command-line window) would appear on a blank background. (The background is provided by `fvwm`, which also manages the keyboard and mouse, as described previously.)

In practice, Linux vendors create complex scripts that check for the availability of default window managers, look at a variety of configuration files, and start numerous X clients to provide a convenient working environment for users. The standard start-up process is outlined in the following list and illustrated in Figure 5-17. Details about the last few items in the list are provided later in this section.

1. The user executes the `startx` command to launch the graphical system.
2. The `startx` command initiates the `xinit` command.
3. The `xinit` command starts the X server program (such as `XFree86_SVGA` or `XFree86_S3`).
4. The `xinit` command attempts to launch a script called `.xinitrc` located in the user's home directory.
5. If the file is found, the commands in `.xinitrc` are executed by `xinit`. This script normally executes commands in other scripts, particularly the `.Xclients` script, which can also be located in each user's home directory, if the system administrator implements a different graphical configuration for each user.
6. If the `.xinitrc` file is not found in the user's home directory, the `xinit` program looks for the file `/etc/X11/xinit/xinitrc`. (Notice that this filename does not begin with a period as do those in a user's home directory.)
7. If the `/etc/X11/xinit/xinitrc` file mentioned in Step 6 is found, the commands in it are executed by `xinit`. The `/etc/X11/xinit/xinitrc` script normally executes commands in other scripts, particularly the `/etc/X11/xinit/Xclients` script. (Again, the filename has no initial period.) These two scripts (located in `/etc/X11/xinit`) are systemwide graphical configuration scripts, which are used for any user who does not have files in his or her home directory as described in Steps 4 and 5.

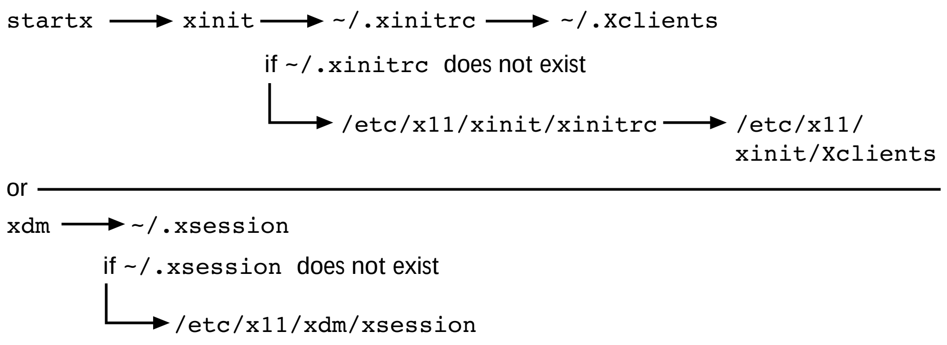


Figure 5-17 The start-up process for X

The scripts `xinitrc`, `.xinitrc`, `Xclients`, and `.Xclients` often include calls to execute other scripts, such as `Xclients_default`. The names and arrangement of these additional scripts vary by Linux distribution, but you can learn more about them by reviewing the contents of the `xinitrc` or `.xinitrc` file. All of the standard scripts (such as `xinitrc`), and the additional scripts that may be used by a particular Linux distribution, are shell scripts, which are discussed in Chapter 12.

Notice in the steps above that employing a separate configuration file in each user's home directory results in a different graphical configuration for each user. One user might choose to use `fvwm` as a window manager; another user might select Gnome or KDE as a graphical environment. A sample `.xinitrc` configuration file is shown here. Notice that the last item executed is a window manager, `twm`.

```
xrdb -load $HOME/.Xresources
xsetroot -solid gray &
xclock -g 50x50-0+0 -bw 0 &
xload -g 50x50-50+0 -bw 0 &
xterm -g 80x24+0+0 -bw 0 &
xterm -g 80x24+0-0 -bw 0 &
twm
```

In Step 3 of the procedure above, the `xinit` program starts the X server. To do this, the `xinit` program looks for the file named `X` in the directory `/usr/X11R6/bin`. This file is a pointer to the actual X server or similar program that helps the system choose an X server. You can see what file it points to by using the `ls` command. For instance, the command `ls -l /usr/X11R6/bin/X` displays an arrow character indicating that the file `X` refers to another file. The file X will refer to either the `XFree86_SVGA`, or the `Xwrapper` file, or another X server program (depending on which X server is needed for the video card). If you use a commercial X server, the file `X` will point to an X server such as `Xaccel` (the Accelerated-X server). This type of pointer is called a symbolic link. (You will learn about symbolic links in Chapter 6.) The sample output of the command `ls -l /usr/X11R6/bin/X` is shown below. You see that the `X` file refers to a file called `Xwrapper`, which is used to launch an X server program.

```
lrwxrwxrwx 1 root root   8 Nov 18 02:45 /usr/X11R6/
     bin/X -> Xwrapper
```

A user's home directory can contain a file named `.xserverrc` that defines which X server to start for that user. All users on a system are likely to employ the same X server, however, because all users on the system rely on the same video card. Thus, you will rarely see this configuration file used.

X Resources

Each graphical application uses a number of separate screen elements, such as scroll bars, text fonts, mouse pointers, and title bars for windows or dialog boxes. Each of these elements is called an **X resource**. This is a term used mostly by programmers; however, you should be familiar with it, because as a system administrator you can configure the appearance of the X resources used in each application.

A collection of default X resource settings applies to all X applications. These default settings govern how windows in the application are displayed, which colors and fonts are used, and which features of the application are active when the application starts. You can also set up additional X resource settings that apply only when a specific user runs a specific application. These resource settings are compiled into a **resource database** file. Collectively, the information in this file defines how an X resource should appear on screen.

The main resource database file is called `app-defaults`. It is located at `/usr/X11R6/lib/X11`. In addition to this set of resource information, each user's home directory can contain additional settings that will override the default appearance of specific applications. A file named `.Xresources` or `.Xdefaults` contains this information. For example, the following sample lines in an `.Xresources` file define how an `xterm` window will appear and what features it will include. These same features can be set or changed once the `xterm` program is running, but when a user creates an X resource database, the application is started with the user's preferences already active.

```
XTerm*cursorColor: gold
XTerm*multiScroll: on
XTerm*jumpScroll: on
XTerm*reverseWrap: on
XTerm*curses: on
XTerm*Font: 6x10
XTerm*scrollBar: on
XTerm*scrollbar*thickness: 5
XTerm*multiClickTime: 500
XTerm*charClass: 33:48,37:48,45-47:48,64:48
XTerm*cutNewline: off
XTerm*cutToBeginningOfLine: off
XTerm*titeInhibit: on
XTerm*ttyModes: intr ^c erase ^? kill ^u XLoad*Background: gold
```

X resource database files are activated within one of the start-up scripts such as `.xinitrc` or `.Xclients` by using the command `xrdb`. The **`xrdb`** command loads an initial X database resource file or adds resource configuration details from files such as `.Xresources`. The following `xrdb` command was included in the resource database given earlier:

```
xrdb -load $HOME/.Xresources
```

When there are additional resource settings to be loaded, the `-merge` option is used instead of the `-load` option. You will rarely need to set up the `xrdb` command, because it is part of the default configuration when you install X. But you may want to add information to an `.Xresources` file to configure the appearance of an application. The online manual page for a graphical application will tell you about options you can configure via X resources.

TIP

Linux desktop environments (such as KDE) include an X resource database that is designed to assign a standard look and feel to all programs that you launch. The desktop environments do not depend on using configuration files such as `.Xresources`. Rather than trying to use an `.Xresources` file to configure the appearance and options of a desktop such as KDE or Gnome, use the configuration tools in these environments.

USING DESKTOP INTERFACES

Not all Linux users want a graphical interface, but for those who do, the latest developments in the field of graphical desktops for Linux provide a number of impressive features. A desktop interface is a graphical environment that provides a collection of functions and utilities to make using the computer easier for those who do not have many commands memorized. You can use a desktop interface to:

- Place icons on the screen's background (where no other windows are visible). Clicking on these icons launches applications or displays data files that the user commonly accesses.
- Manage multiple applications efficiently using toolbars, keyboard controls, and so forth.
- Use menus to access frequently used utilities and applications. You can customize these menus to meet your specific requirements.
- Use a collection of basic applications provided along with the desktop, such as a text editor, calculator, calendar, note-taking application, music player, and so forth.
- Use a convenient, integrated file management utility to view and manipulate files.
- Use applications created specifically for the desktop interface. These applications rely on a common set of functions that create a single look and feel for the environment, reduce the system resources required (because of shared functionality), and interact with other applications via drag-and-drop features (or some other type of application linking feature).

Powerful window managers such as `fvwm` include some, but not all, of the features discussed above. If you encounter all of these features on an `fvwm`-based system, it's because your Linux vendor or system administrator has combined numerous Linux applications into a working graphical environment. But this cobbling together of pieces does not result in an integrated environment in which applications have the same look and feel, can share data, and can be easily managed from a central set of desktop utilities. A complete desktop interface does provide all of these benefits. The advent of complete desktop interfaces is likely to be a primary reason for increased use of Linux in the coming years.

The KDE Interface

In 1996 Matthias Ettrich began creating a full-featured graphical environment for Linux. He dubbed the project the K Desktop Environment (KDE). KDE is now the most widely used graphical environment on Linux systems. When you install most Linux distributions, KDE is installed by default with a complete set of KDE applications. KDE applications use the same graphical toolkit as the desktop itself and can thus share functionality such as common dialog boxes and drag-and-drop capability.

KDE includes a suite of applications for Internet access, system maintenance, personal productivity (organizers, calculators, music players), and many other basic tasks. KDE includes a set of icons at the bottom of the screen known as the Panel. You can use the icons in the Panel to start common applications such as Netscape Communicator or a terminal window. Each application that you start in KDE appears as a button in the taskbar (which is normally at the top of the KDE screen). You can switch between applications or open a minimized application using the corresponding button on the taskbar. A few standard icons are included on the KDE default desktop; each user can add others. Figure 5-18 shows a basic KDE desktop.

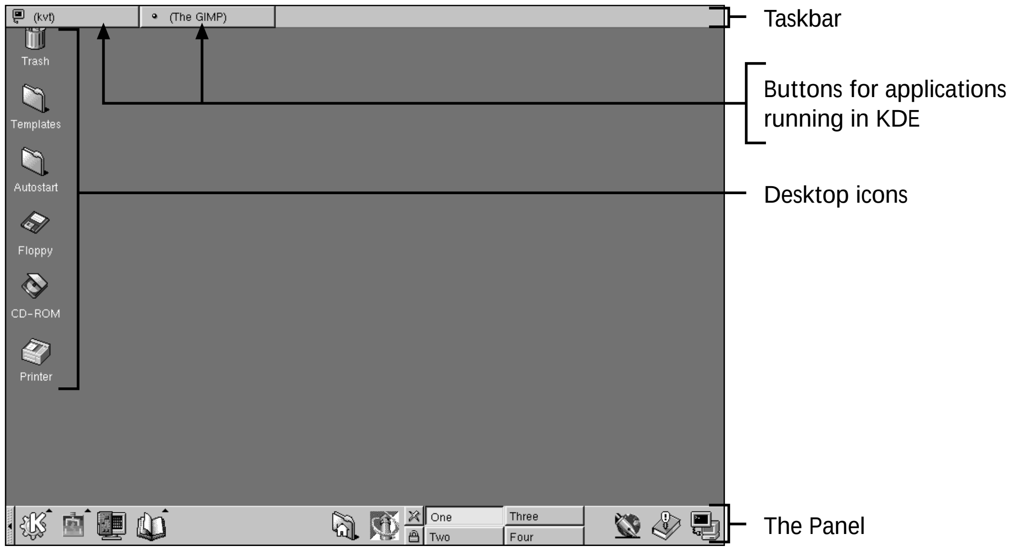
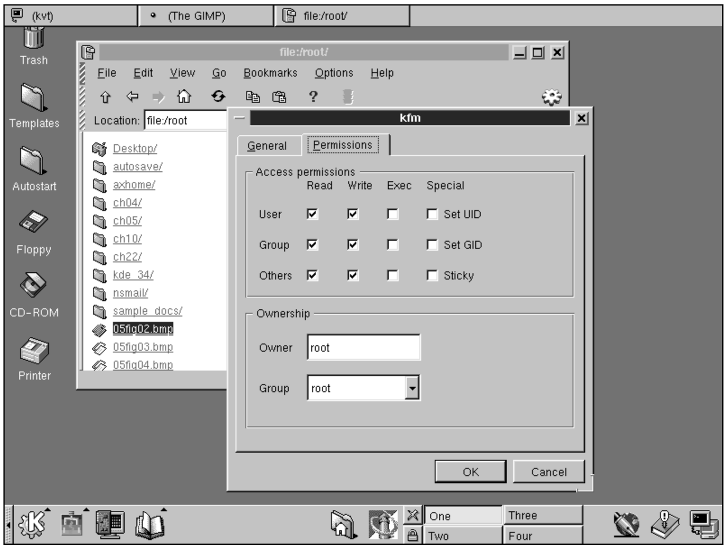


Figure 5-18 The KDE Desktop interface

KDE includes a powerful file manager program called `kfm` that provides several views of files (such as an icon view and a detailed view) and also doubles as a Web browser (much like the Microsoft Explorer program). The file manager in KDE is designed to assign icons to files automatically, based on the file's type (such as text file, program, or graphics file). Based on the type of file, the file manager also assigns default capabilities that allow a user to perform appropriate actions. For example, you will be able to click on a text file to view that file, or you can drag and drop a graphics file onto a desktop icon to view that graphics file in the program represented by the icon. You can right-click on a file and choose Properties from the pop-up menu to open a dialog box in which you can adjust the file's properties (including the file's name and access rights). Figure 5-19 shows a file manager window with a dialog box showing properties for one file.



Location: file:/root

Access permissions

	Read	Write	Exec	Special
User	✓	✓	☐	☐ Set UID
Group	✓	✓	☐	☐ Set GID
Others	✓	✓	☐	☐ Sticky

Ownership

Owner: root

Group: root

Figure 5-19 The KDE file manager window, with properties for one file

KDE includes several system administration utilities (described in later chapters), which you can use to manage your Linux system. At the same time, KDE includes a complete Control Center in which you can configure KDE itself. Via the Control Center, you can configure color schemes, screen savers, fonts, languages used (KDE supports over 30 languages, including Chinese and Russian), and the position of each part of the KDE screen (such as where the Panel and taskbar are positioned). Advanced options in the Control Center let you decide what information should be included on the title bar of each application window and which keystrokes should switch between applications. Many other options are available. Figure 5-20 shows the KDE Control Center with some of the configuration options for desktop appearance. Each user's configuration choices are stored separately.

The left side of the KDE Control Center window displays a list of sections, such as `Desktop`, `Applications`, and `Network`. By clicking the plus sign next to any section name, you will see a list of items within that section. Clicking on one of these items displays configuration options on the right side of the Control Center window.

Although a default set of desktop icons is included with KDE, each user can create additional icons, assign icons to applications, and add items to menus as desired. Applications created for the KDE Desktop can interact via drag-and-drop and other methods. Non-KDE graphical applications cannot interact in this way. Although non-KDE applications cannot use all KDE features (such as drag-and-drop), you can add any application to a KDE menu or assign it to an icon on the KDE Desktop.

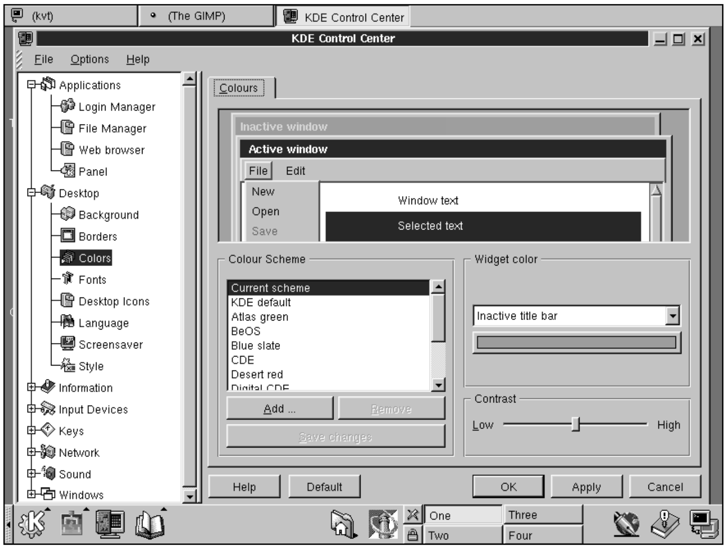


Figure 5-20 The KDE Control Center

Development of KDE continues, pushed forward by a worldwide staff of volunteer programmers, designers, writers, and translators. To learn more about the KDE project, visit the Web site *www.kde.org*.

When KDE is installed on a Linux system, the KDE components are integrated into the standard X Window System configuration files, so that the KDE window manager (called `kwm`) and related KDE programs are launched when you execute the `startx` command. Some distributions create special commands that you should use to start KDE. Caldera OpenLinux uses the command `kde` to start the KDE desktop. (The `startx` command in OpenLinux starts a simple `fvwm`-based display.) In Red Hat Linux, you can use the `switchdesk-kde` utility to select KDE as the default desktop environment.

The standard KDE installation places all KDE files in the directory `/opt/kde`. Some systems (Red Hat Linux) place KDE in a different directory, such as `/usr`. In addition, the home directory of each user who runs KDE contains a subdirectory named `.kde`. This subdirectory contains configuration files specific to that user for any KDE applications the user has reconfigured (that is, the user has selected settings other than the default KDE settings). You can alter the KDE configuration using the standard KDE graphical tools such as the KDE Control Center, although advanced KDE users may choose to edit the configuration files directly.

The Gnome Desktop

At about the time that the KDE project was first becoming popular, a group of developers within the Linux community became concerned about the fact that KDE was based on a graphical library that was not licensed using the GPL. Troll Tech, the makers of the Qt graphical library used by KDE, permitted the graphical library to be freely distributed with KDE, but the license for commercial users was stricter than the GPL used by Linux itself. Because of this concern, a new desktop project was founded. This project, the Gnu Object Model Environment (GNOME, usually written as Gnome) resulted in a complete desktop environment similar to KDE, but governed by licensing terms of the GPL.

TIP Because the Gnome project is associated with the GNU project, the name of the desktop interface is pronounced with a hard letter *g*. Pronounce it as "Gh-nome," like GNU.

While Gnome was being developed, Troll Tech decided to change its licensing agreement to mirror the terms of the GPL. Thus, Linux now boasts two completely free, open source desktop environments.

Gnome is very similar to KDE. It includes a Panel with an integrated taskbar (containing a button for each running application). Icons on the desktop let you launch commonly used applications or data files. A main menu provides quick access to dozens of applications written to take advantage of the look and functionality of Gnome. The applications included with Gnome are similar to those provided with KDE. They include a powerful file manager, personal productivity applications, and system maintenance utilities. Figure 5-21 shows the Gnome Desktop with a file manager window and the main menu open for viewing.

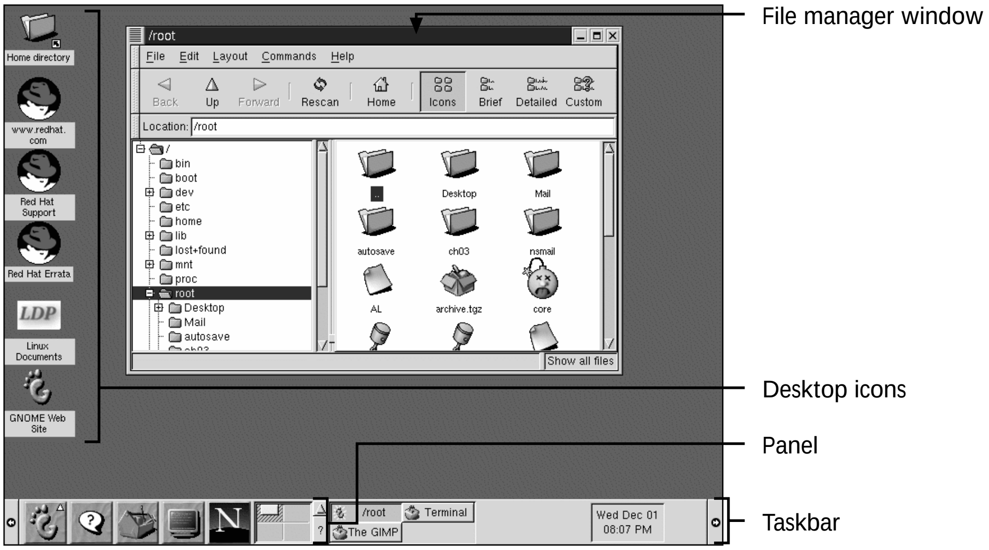


Figure 5-21 The Gnome Desktop with a file manager window

From the beginning, the Gnome project received encouragement and financial support from Red Hat Software. As a result, Red Hat is currently the only major Linux distribution that includes Gnome. Granted, Red Hat is the largest Linux distributor, so Gnome enjoys a large following of supporters. But the Gnome Desktop is not as far along in development (and hence is not as stable or complete) as the KDE Desktop.

Gnome includes a Control Center similar to the one in KDE, in which you can set up the appearance of the desktop. This tool is shown in Figure 5-22.

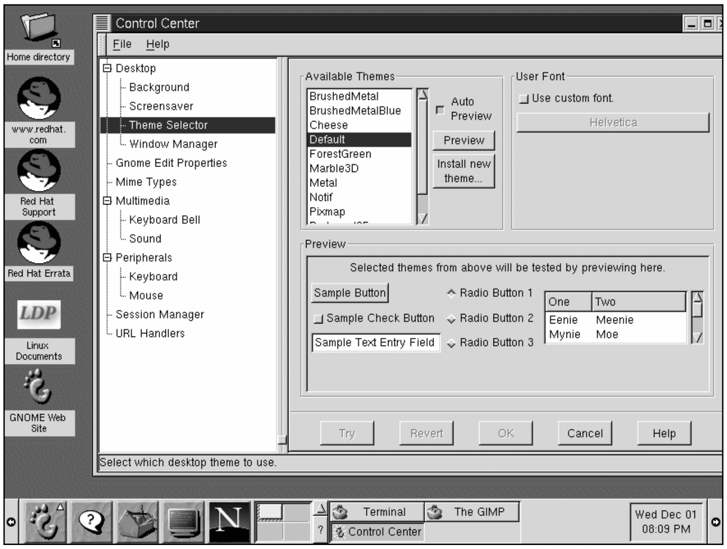


Figure 5-22 The Gnome Control Center

Although Red Hat Linux favors the Gnome Desktop, Red Hat also recognizes the prominence and features of KDE. The Red Hat Linux installation provides an option to select KDE as the default desktop, or to install KDE in addition to Gnome. When KDE is installed as part of a Red Hat Linux system, you can execute the `switchdesk-kde` program to open a dialog box in which you can choose the desktop interface you prefer to use: Gnome or KDE.

USING A GRAPHICAL LOGIN SCREEN

If the installation of Linux is successful, most Linux distributions will start the system in runlevel 5 rather than runlevel 3. As explained in Chapter 4, runlevel 5 is equivalent to the standard operating mode (runlevel 3) except that runlevel 5 always provides a graphical interface. After Linux starts and initializes, it switches immediately to graphics mode. A graphical login screen similar to the one shown in Figure 5-23 appears. This screen is equivalent to the character-mode login screen in that it requires you to enter a valid username and password before using the system. Once the username and password have been entered, a standard graphical system such as `fvwm`, Gnome, or KDE is launched.

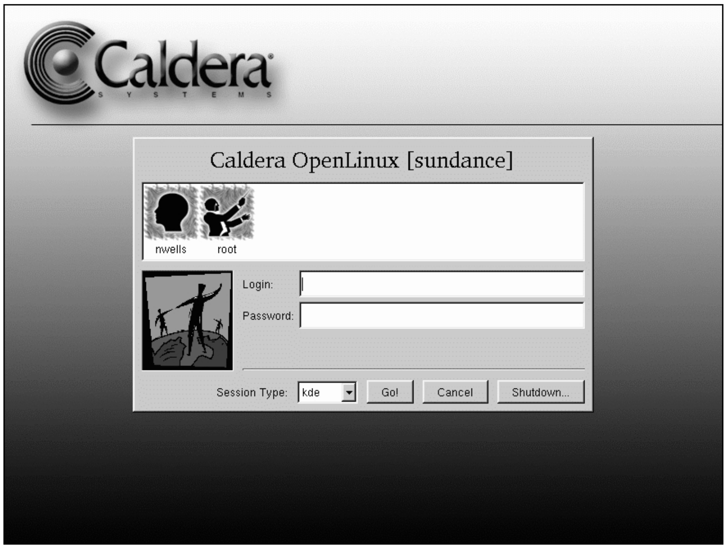


Figure 5-23 Graphical login screen

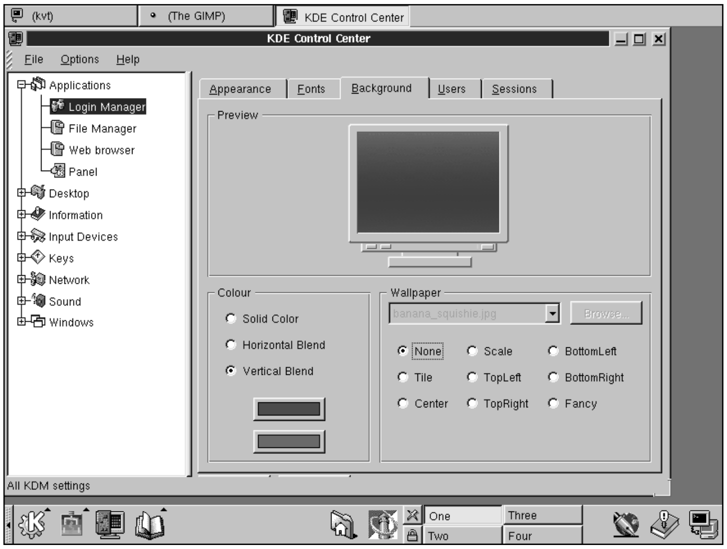
TIP To change the runlevel used by default as the Linux system starts, set the value in the `initdefault` line of the `/etc/inittab` file, as outlined in Chapter 4.

The graphical login screen shown in runlevel 5 is provided by the X display manager, or `xdm` program. `Xdm` is started by the `init` program as Linux boots. Any time a user exits the graphical environment, `xdm` is restarted automatically to provide a graphical login screen. Thus the user never encounters a character-mode screen—a fact that can make new users much more comfortable with Linux. The `xdm` program is installed as part of the X Window System.

When a user logs in using an `xdm` graphical login screen, `xdm` selects which programs to start based on the session chosen by the user. A **session** defines a set of graphical programs to run when a user logs in. Sessions are often named after window manager or desktop choices, such as KDE, `fvwm`, or Gnome. Another common session name is `failsafe`. The `failsafe` session opens a single terminal window and is used by the `root` user when troubleshooting the system.

A configuration file called `xsession` specifies which programs are started by a particular session name. When `xdm` starts X (after a user logs in), it executes the file `/etc/x11/xdm/Xsession` to determine which X clients to launch. The `Xsession` file contains a set of instructions that match the names of sessions defined for `xdm`. For example, if the user selects the `failsafe` session, the `Xsession` script will only launch an `xterm` program; if the user selects a KDE session, `Xsession` starts all of the standard KDE components. A per-user configuration file named `.xsession` (all lowercase, beginning with a period) can be placed in a user's home directory to control which sessions that user has available. The available session types are normally common to all users, however, so using an `.xsession` file in a home directory is unusual.

Different versions of `xdm` have been created for different environments. For example, the KDE Desktop uses a program called `kdm` (the KDE display manager). The `kdm` program is basically identical to `xdm` in functionality, but `kdm` can be graphically configured within KDE. The `xdm` program must be configured using a series of complex options within the files of the `/etc/X11/xdm` directory. Figure 5-24 shows a sample configuration screen for `kdm` within the KDE Control Center. (The `kdm` program is configured by the `Login Manager` item in the `Applications` section of the Control Center.)



Colour: [✓ Vertical Blend]
Wallpaper: banana_squishie.jpg [✓ None]

Figure 5-24 Configuring `kdm` in KDE

Similarly, other Linux distributions provide graphical tools you can use to configure `xdm` (or a similar program supplied by the Linux vendor as a graphical login display manager). The `LinuxConf` utility in Red Hat Linux, for example, provides this capability.

CHAPTER SUMMARY

- The X Window System is a powerful and flexible graphical system. A window manager or desktop interface is required to provide a user interface to X. Many different utilities can be used to configure an X server, but it remains a challenging process.
- The X Window System is launched by the `startx` command. This command relies on a number of scripts to control which graphical programs are started, including which

window manager or desktop interface is used. By editing these scripts, or configuration files, a user controls how the graphical environment appears.

- Desktop interfaces include many features to promote ease of use in Linux, such as menus, toolbars, desktop icons, and convenient file management utilities. The two primary desktop interfaces on Linux are KDE—which is used on almost all Linux distributions in the world—and Gnome, which is included with Red Hat Linux, the most widely used Linux distribution in the world. The two desktop interfaces are quite similar in design and functionality.
- A graphical login screen is provided in Linux by the `xdm` program. This program uses a configuration file containing session definitions to determine what programs to launch after a user has entered a valid username and password at the graphical login screen.

KEY TERMS

background application — An application that does not stop the program that started it from going on to other tasks.

desktop environment — A graphical application that provides a comprehensive interface, including system menus, desktop icons, and the ability to easily manage files and launch applications.

graphical libraries — Collections of programming functions that an X client can use to more efficiently create and manage the elements of a graphical environment.

`lizardx` — A graphical utility used for configuring the X Window System.

Project Athena — The project sponsored by DEC and MIT to create a graphical environment or windowing system for UNIX.

resource database — A file that defines how an X resource should appear on screen.

session — A configuration that defines a set of graphical programs to run when a user logs in.

window manager — A special-purpose graphical application (X client) that controls the position and manipulation of the windows within a graphical user interface.

X client — A graphical application.

X resource — The separate screen elements of a graphical application, such as scroll bars, text fonts, mouse pointers, and title bars for windows or dialog boxes.

X server — The program that communicates with the video card to create images on the screen.

`Xconfigurator` — A utility in Red Hat Linux for configuring the X Window System.

`xf86config` — A standard text-based utility for configuring the X Window System.

`XF86Setup` — A graphical utility for configuring the X Window System.

`xrdb` — A command that loads an initial X database resource file or merges additional resource configuration details.

REVIEW QUESTIONS

1. The historical beginnings of the X Window System originated with:
   a. Project Athena
   b. Early versions of Microsoft Windows
   c. The Apple Macintosh computer
   d. The XFree86 Project
2. The X Window System was released under the same license (the GPL) as Linux. True or False?
3. Describe the function of an X server within the X Window System.
4. A(n) ________________ is another name for any graphical application running in the X Window System.
   a. window manager
   b. X client
   c. `xinit`
   d. X resource
5. Name four window managers that can be used with the X Window System on Linux.
6. A window manager is best described as:
   a. A special-purpose X client that provides core graphical functionality for other X clients
   b. The management tool that administers the underlying X server
   c. The program that communicates directly with a video card in order to create windows on a computer's screen
   d. An optional component used to improve the appearance of some graphical applications
7. Name the two graphical libraries used by the two major desktop environments of Linux.
8. A graphical library is best described as:
   a. The functions that control the video card
   b. A collection of functions that any graphical program can use to create a common look and feel
   c. The configuration program used to set up X
   d. The X client that draws all windows and handles mouse and keyboard input
9. To view an X application remotely on a Windows-based computer, you must add a(n) ________________ to the Windows computer.
   a. X client
   b. `xinit` program
   c. X server software
   d. Qt library package

10. The X Window System files are normally located in which subdirectory in Linux:
    a. `/x`
    b. `/usr/graphics`
    c. `/opt/kde`
    d. `/usr/X11R6`
11. On any Linux system you can choose not to run the X Window System in order to conserve system resources. True or False?
12. The `Xconfigurator` program is a standard X configuration tool in which Linux distribution?
    a. SuSE Linux
    b. TurboLinux
    c. Debian Linux
    d. Red Hat Linux
13. Name four programs that you can use to configure the XFree86 X server programs.
14. Explain why configuring the X server is challenging, but configuring other components of X is fairly straightforward.
15. The ________________ program is considered the easiest, most effective tool for configuring XFree86 servers.
    a. `lizardx`
    b. `xf86config`
    c. `XF86Setup`
    d. `xdm`
16. Which is *not* considered during a standard X configuration process?
    a. The mouse
    b. The video card
    c. The graphical fonts used
    d. The screen resolution
17. Besides the standard configuration programs available for Linux, describe four other methods of obtaining information to help you configure X.
18. The ________________ command is normally used to launch the X Window System.
    a. `xinit`
    b. `startx`
    c. `xdm`
    d. `xinitrc`

19. The `xinit` program searches for a variety of configuration files, including which of the following?
    a. `/etc/startx`
    b. `/etc/X11/xinit/xinitrc`
    c. `/opt/kde/share/config/kdmrc`
    d. `/tmp/xsession`
20. Explain the steps used by `xinit` to launch X, name four configuration files that `xinit` checks for, and describe the types of alterations or additions that a Linux distribution might make to the default steps.
21. To function correctly, the `startx` and `xinit` programs require that:
    a. A symbolic link named `/usr/X11R6/bin/X` is created to point to an X server program
    b. The `xdm` program has been properly configured
    c. All of the X client scripts have the 644 file permission assigned
    d. X resources have not been modified by the root user
22. Name four features associated with a modern desktop environment.
23. Describe the history of the Gnome project as it relates to the KDE project, including naming the graphical library on which each project is based.
24. The `xrdb` command is used to manage:
    a. X resources using X resource database files
    b. Automatic execution of X applications
    c. The configuration of a screen saver application
    d. The display manager `xdm`
25. The `xdm` program is started by the ________________ program (or script).
    a. `init`
    b. `startx`
    c. `xinit`
    d. `xrdb`
26. A session used in `xdm` refers to:
    a. A set of applications to be started, as defined by the system administrator or Linux vendor
    b. One login and logout event by a user
    c. The time spent running a single graphical application
    d. The resources assigned to X by the Linux system

27. An `xdm` session is normally defined by information in:
    a. An `Xsession` file
    b. The resource database
    c. The KDE login manager screen
    d. An `Xclients` file

HANDS-ON PROJECTS

Hands-On Project

Project 5-1

In this activity you search the Web for information on video card compatibility. To complete this activity you should have a computer with a Web browser and an Internet connection.

1. Go to the XFree86 Project Web page at **www.xfree86.org**.
2. Scroll to the bottom of the page, reviewing the information on the page as you do so. Near the bottom of the page, in the section on XFree86 documentation, locate the link to the XFree86 video card/server list. Click on that link.
3. Review the list of video cards. What does the X server name on the right side of the list correspond to? How would you make use of this information or check that it was used correctly on the Linux system you are configuring? What steps are recommended at the top of this Web page if the video card you are searching for is not listed?
4. Check another source of information by entering the following URL in your Web browser: **www.calderasystems.com**. When the Web page appears, click the **OpenLinux 2.3** link.
5. When the OpenLinux 2.3 page appears, click the **Hardware Compatibility** link. A list of compatible hardware appears.
6. Click the **Video** link. What is this link titled? Although you are viewing the Caldera Systems Web site, what information do you see that would make you comfortable using this list for other Linux systems?
7. When a list of compatible video cards appears, review its contents. Why do you think this list is longer than the list of cards at the XFree86 Project Web site? How could the information at the bottom of the page about additional X servers help you? How would you make use of it?
8. Suppose you needed to configure an Intel Real3D Starfighter AGP card with an i740 video chipset. Click the link at the bottom of the page that would tell you more about how to support that video card.
9. Go to the Linux Documentation Project site at **www.metalab.unc.edu/LDP**.
10. Click **HOWTO** at the top of the page. A list of available formats for the HOWTO documents is presented. Click **plaintext**.

11. An index of HOWTO documents appears. Scroll down and click **Hardware-HOWTO**. This large file takes a few moments to load. Scroll down to section 6, which is titled "Video Cards." Does this information resemble the information on the Caldera Systems Web site?

Project 5-2

In this activity you continue researching video card information using commercial X server products. To complete this activity you should have a computer with a Web browser and an Internet connection.

1. Go to the Xi Graphics site at **www.xig.com**.
2. Click **Hardware Support Library** at the top of the page. A hardware information page appears. Click the link leading to the list of supported graphics cards.
3. After a moment a list of card manufacturers appears. Scroll down, reviewing the list.
4. Review the information for the Intel chips supported by the Xi Graphics product. Based on what you learned in Project 5-1, do you think the Xi Graphics product will support the Starfighter video card?
5. Click on the link to display benchmark data. Review the information in the Web page that appears. Why might you choose to purchase a commercial X server from Xi Graphics rather than use the free X servers included with a Linux distribution?
6. Go to the MetroLink home page at **www.metrolink.com**. Click the **Products** link. The Products page for the MetroLink X servers (and other related products) appears.
7. Click the link that displays video cards supported by the Metro-X product.
8. Review the list of supported graphics cards that appears. If the list seems shorter than the XFree86 and Xi Graphics lists, explore the MetroLink Web site to determine why a person might purchase the MetroLink products rather than use the XFree86 servers. Are some graphics cards listed on the MetroLink site that are not listed on the other two sites?

Project 5-3

In this activity you explore the X Window System configuration files that are included in a Linux system. To complete this activity you must have an installed Linux system with a valid user account (root access is not required, however).

1. Log in to the Linux system using a valid username and password.
2. If you logged in using a graphical login screen and are now viewing a graphical desktop, open the **`main`** menu of the desktop, and then choose **`Terminal`** from the `Utilities` submenu. A terminal window opens, where you can enter commands.
3. Change to the `/etc/X11/xinit` directory with this command:
   `cd /etc/X11/xinit`.
4. Enter **`ls`** to list the files in this directory. What files do you see that were discussed in the chapter? Can you describe which user or users will use these files? What additional information do you need to determine whether a user will access these files when starting X?

5. Change to your home directory by entering the **`cd`** command without any parameters.
6. List all the files in your home directory, including hidden files, by entering the command **`ls -a`**.
7. What files can you see that were mentioned in the chapter? Do you see additional files that appear to be related to the X Window System?
8. If you see an `.Xdefaults` file, review its contents using the command **`cat .Xdefaults`** (include the period at the beginning of the filename). What effects do the contents of this file have when you log in and start X? Could you add information to this file if you had proper documentation on an X application?
9. Change to the `xdm` directory using the command **`cd /etc/X11/xdm`**.
10. List the files in this directory with the **`ls`** command. Do you see a file in which the sessions available on the system are defined?
11. Use the following `more` command to view the `Xsession` file a page at a time: **`more Xsession`**. Can you identify any familiar items in this script file, such as the name of a window manager or desktop environment? (Remember that this file is only used if you are using a graphical login screen.)

Case Projects

1. The Starwood movie studio has decided to create a movie using computer animation and wants to use Linux on the computers to help with some of the graphical work required. You have been asked to help set up the Linux graphics workstations that the studio will use. As you might expect, the studio is in a big hurry to finish the production. Fortunately, money is not a limiting factor in your decisions. The studio wants to use a new graphics card that provides high-resolution images and high-speed preparation of computer animation. You are concerned that the card is not yet supported on Linux versions of the X Window System. How will you determine whether the card is supported? If the standard XFree86 software does not support the card, what steps could you take to locate or create support among those who work with X and Linux?
2. Once you have installed Linux and configured the new video cards successfully, the studio asks you to adjust the features of the system to display a nonstandard resolution of 1420 × 865 pixels. You know that X is flexible enough to do this, but the configuration tools don't provide this screen resolution as an option. How will you learn more about the video card capabilities and X configuration in order to grant this odd request?
3. As the movie winds down production, and all the graphics work is done, the studio asks you to prepare the Linux systems to be used as regular desktop workstations in the studio. How will you configure the graphical portion of Linux to be used by the inexperienced workers in that section of the studio? What components of X will you include? How will you seek to make the systems easy to use for common daily computing tasks?